INFORMATION TECHNOLOGY, COMPUTER SCIENCE AND MANAGEMENT
ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

Check for updates

UDC 004.4
https://doi.org/10.23947/2687-1653-2023-23-1-76-84

Original article

Data Warehouse Failover Cluster for Analytical Queries in Banking

Victor V Sivov ✉, **Vladimir A Bogatyrev**
ITMO University,49, Kronverksky Pr., St. Petersburg, Russian Federation

✉ v.sivov777@gmail.com

Abstract

Introduction. The banking sector assigns high priority to data storage, as it is a critical aspect of business operations. The volume of data in this area is steadily growing. With the increasing volume of data that needs to be stored, processed and analyzed, it is critically important to select a suitable data storage solution and develop the required architecture. The presented research is aimed at filling the gap in the existing knowledge of the data base management system (DBMS) suitable for the banking sector, as well as to suggest ways for a fault-tolerant data storage cluster. The purpose of the work is to analyze the key DBMS for analytical queries, determine the priorities of the DBMS for the banking sector, and develop a fault-tolerant data storage cluster. To meet the performance and scalability requirements, a data storage solution with a fault-tolerant architecture that meets the requirements of the banking sector has been proposed.

Materials and Methods. Domain analysis allowed us to create a set of characteristics that a DBMS for analytical queries (OnLine Analytical processing — OLAP) should correspond to, compare some popular DBMS OLAP, and offer a fault-tolerant cluster configuration written in xml, supported by the ClickHouse DBMS. Automation was done using Ansible Playbook. It was integrated with the Gitlab version control system and Jinja templates. Thus, rapid deployment of the configuration on all nodes of the cluster was achieved.

Results. For OLAP databases, criteria were developed and several popular systems were compared. As a result, a reliable cluster configuration that met the requirements of analytical queries has been proposed for the banking industry. To increase the reliability and scalability of the DBMS, the deployment process was automated. Detailed diagrams of the cluster configuration were also provided.

Discussion and Conclusions. The compiled criteria for the DBMS OLAP allowed us to determine the need for this solution in the organization. Comparison of popular DBMS can be used by organizations to minimize costs when selecting a solution. The proposed configuration of the data warehouse cluster for analytical queries in the banking sector will improve the reliability of the DBMS and meet the requirements for subsequent scalability. Automation of cluster deployment by the mechanism of templating configuration files in Ansible Playbook provides configuring a ready-made cluster on new servers in minutes.

Keywords: DBMS, OLAP, data warehouse, ClickHouse, failover cluster.

Acknowledgements. The author would like to thank V.A. Bogatyrev, Dr.Sci. (Engineering), professor of the Computer Engineering Department of ITMO University, Honorary Worker of Science and Technology of the Russian Federation, who conducted expert interviews together with the authors of the article.

For citation. Sivov VV, Bogatyrev VA. Data Warehouse Failover Cluster for Analytical Queries in Banking. *Advanced Engineering Research (Rostov-on-Don)*. 2023;23(1):76–84. https://doi.org/10.23947/2687-1653-2023-23-1-76-84

Научная статья

Отказоустойчивый кластер хранилища данных для аналитических запросов в банковской сфере

В.В. Сивов✉, **В.А. Богатырев**

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Российская Федерация, г. Санкт-Петербург, Кронверкский пр., д. 49

✉ v.sivov777@gmail.com

Аннотация

Введение. Банковский сектор придает большое значение хранению данных, поскольку это критически важный аспект бизнес-операций. Объем данных в данной сфере неуклонно растет. С увеличением объемов данных, которые необходимо хранить, обрабатывать и анализировать, крайне важно выбрать подходящее решение для хранения данных и разработать необходимую архитектуру. Представленное исследование направлено на то, чтобы заполнить пробел в существующих знаниях СУБД, подходящих для банковского сектора, а также предложить способы для отказоустойчивого кластера хранения данных. Цель работы — анализ ключевых СУБД для аналитических запросов, определение приоритетов СУБД для банковского сектора и разработка отказоустойчивого кластера хранения данных. Для выполнения требований к производительности и масштабируемости предложено решение для хранения данных с отказоустойчивой архитектурой, отвечающее требованиям банковского сектора.

Материалы и методы. Анализ предметной области позволил создать набор характеристик, которым должна соответствовать СУБД для аналитических запросов (OLAP), выполнить сравнение некоторых популярных OLAP СУБД и предложить отказоустойчивую кластерную конфигурацию, написанную на языке xml, поддерживаемую СУБД ClickHouse. Автоматизация выполнена с помощью Ansible Playbooks. Он интегрирован с системой управления версиями Gitlab и шаблонами Jinja. Таким образом достигается быстрое развертывание конфигурации на всех нодах кластера.

Результаты исследования. Для баз данных OLAP были разработаны критерии, проведен сравнительный анализ нескольких популярных систем. В результате была предложена надежная кластерная конфигурация в банковской индустрии, которая удовлетворяет требованиям аналитических запросов. Для увеличения надежности и масштабируемости СУБД процесс развертывания был автоматизирован. Также приведены детальные схемы конфигурации кластера.

Обсуждение и заключения. Составленные критерии для OLAP СУБД позволяют определить необходимость данного решения в организации. Сравнение популярных СУБД может быть использовано организациями для минимизации затрат при выборе решения. Предлагаемая конфигурация кластера хранилища данных для аналитических запросов в банковской сфере позволит повысить надежность СУБД и удовлетворить требования к последующей масштабируемости. Автоматизация развертывания кластера путем механизма шаблонизации конфигурационных файлов в Ansible Playbooks позволяет настроить готовый кластер на новых серверах за минуты.

Ключевые слова: СУБД, OLAP, хранилище данных, ClickHouse, отказоустойчивый кластер.

Благодарности. Автор выражает благодарность В.А. Богатыреву, доктору технических наук, профессору кафедры вычислительной техники Университета ИТМО, почетному работнику науки и техники РФ, проводившему экспертные интервью совместно с автором статьи.

Для цитирования. Сивов В.В., Богатырев В.А. Отказоустойчивый кластер хранилища данных для аналитических запросов в банковской сфере. *Advanced Engineering Research (Rostov-on-Don).* 2023;23(1):76–84. https://doi.org/10.23947/2687-1653-2023-23-1-76-84

Introduction. Data storage in the banking sector is one of the key business factors. To ensure the security of customer information and transactions, it is required to take measures of protection, distribution and creation of backups. For operational analysis, employees should be able to make operational analytical requests to the data warehouse, while not interfering with the work of other processes within the organization and without causing a heavy load on the storage itself. Databases and Data Warehouse are information systems in which data is stored, but they are also used to solve various tasks. The article describes what such systems do, what the main differences between them are, and why their effective use is essential for business development.

Many organizations make mistakes in designing the architecture of databases and data warehouses, losing sight of aspects of information security, scalability and fault tolerance. The urgency of this problem is due to the intensive development of systems in banks, the expansion of their fields of application and the increase in the amount of data in need of constant analysis. For operational analysis of a large amount of data, a storage is needed that must meet all reliability and security requirements.

Effective decision-making processes in business depend on high-quality information. In today's competitive business environment, flexible access to a data warehouse is required, organized in such a way as to increase business productivity, provide fast, accurate and up-to-date data understanding. The data warehouse architecture is designed to meet such requirements and is the basis of these processes [1–5].

The objective of the work is to determine the priority DBMS for performing analytical queries in the banking sector and design a fault-tolerant data warehouse cluster. This solution will significantly increase the speed of execution of analytical queries, solve problems with scalability and reliability of the data warehouse.

Materials and Methods. The database stores real-time information about one specific part of the business. Its main task is to process daily transactions. Databases use Online Transaction Processing (OLTP) to quickly delete, insert, replace and update a large number of short online transactions.

Data warehouse is a system that collects data from lots of different sources within an organization for reporting and analysis, using operational analytical processing (OLAP) to quickly analyze large amounts of data. This system focuses on reading, rather than changing historical data from lots of different sources, therefore, compliance with ACID (Atomic, Consistent, Isolated and Durable) requirements is less strict. Data warehouses perform complex functions of aggregation, analysis and comparison of data to support management decision-making in companies.

A warehouse in the banking sector may contain:

– user account information (personal data, addresses, phone numbers);

– information about banking products and services (loans, deposits, plastic cards, mobile banking, etc.);

– data on transactions (including card transactions) in minimal detail for the last three years;

– information about accounts, balances on them, etc.

To meet the needs for OLAP, there are separate types of database management systems (DBMS) [3–6]. Each of the systems has its own characteristics in the construction of architecture.

To perform an effective analysis of compliance with these requirements, warehouses must:

– have a high capacity capable of accommodating huge amounts of data (billions or trillions of rows);

– be organized as wide tables with multiple columns;

– perform queries with a small number of columns;

– have a high query execution speed (in milliseconds or seconds);

– provide for most of the read-only requests;

– support fast bulk data loading when updating (more than 1,000 rows at a time) and adding, but without changing them;

– have high throughput to process a single request (up to billions of rows);

– have high reliability;

– ensure data security and consistency.

For the OLAP scenario of work in the banking sector, it is preferable to use column-based analytical databases, since they can store a lot of columns in a table, which will not affect the speed of reading data. Column-based DBMS provide strong compression of data in columns, since data in one column of the table is usually of the same type, which cannot be said about a row. They also enable to get a tenfold increase in query execution speed on lower-power equipment. At the same time, thanks to compression, the data will occupy 5-10 times less space on the disk than in the case of traditional DBMS [7–11].

During the requirements analysis, the following column DBMS were selected: ClickHouse, Vertica, Amazon Redshift.

ClickHouse is the preferred solution due to the following advantages: open source; it is possible to define some or all structures that will be stored only in memory; high speed; good data compression; http and command line interface; cluster can be scaled horizontally; high availability; ease of installation and configuration. Installation is carried out on the organization's servers in an isolated segment, which meets the security requirements for sensitive data in the banking sector. The DBMS is also included in the register of domestic software; therefore, it provides implementing this software product in state-owned companies.

Amazon Redshift solution is provided only as a cloud service. For organizations from the banking sector that cannot place their data in the clouds for a number of security-related reasons, this product loses its appeal.

Vertica is an alternative version of ClickHouse with a paid license for large clusters and the installability on the company's local servers.

The implementation of the distributed data warehouse architecture is presented below. To increase fault tolerance and performance, the implementation of a distributed ClickHouse failover cluster with three shards and two replicas is proposed.

Sharding (horizontal scaling) makes it possible to write and store parts of data in a distributed cluster, process and read them in parallel on all nodes of the cluster, increasing data throughput.

Replication is copying data to multiple servers; thus, each bit of data can be found on multiple nodes.

Scalability is determined by sharding or segmentation of data. The reliability of the data warehouse is determined by data replication [12–16].

Sharding and replication are completely independent, different processes are responsible for them. It is required to localize small data sets on one shard and ensure a fairly even distribution across different shards in the cluster. To do this, it is recommended to take the hash function value from a field in the table as a sharding key.

Sharding and replication are completely independent, different processes are responsible for them. It is required to localize small data sets on one shard and ensure a fairly even distribution across different shards in the cluster. To do this, it is recommended to take the hash function value from a field in the table as a sharding key.

Depending on the number of available resources and servers, it is proposed to implement this configuration on 3 or 6 nodes. For a production environment, it is recommended to use a cluster of 6 nodes. It should be noted that replication does not depend on sharding mechanisms and works at the level of individual tables, and also, since the replication coefficient is 2, each shard is represented in 2 nodes [17–19]. Configuration options are described below.

The logical topology diagram is as follows:

3(Shard) × 2(Replicas) = Clickhouse Cluster of 6 nodes.

The probability of trouble-free operation of a system with 2 replicas and 3 shards on 6 nodes is equal to:

$$P_c = [1-(1-p)^2]^3.$$

The probability of trouble-free operation is an objective possibility that the system will work for time t without restorations [7, 13].

Thus, a table containing 30 million rows will be distributed evenly across 3 nodes of the cluster. The remaining 3 nodes will store replicas of the data. When one of the cluster nodes is disabled, data will be taken from another available node that contains its replica, thereby achieving reliability [20]. A cluster of 6 nodes is shown in Figure 1.

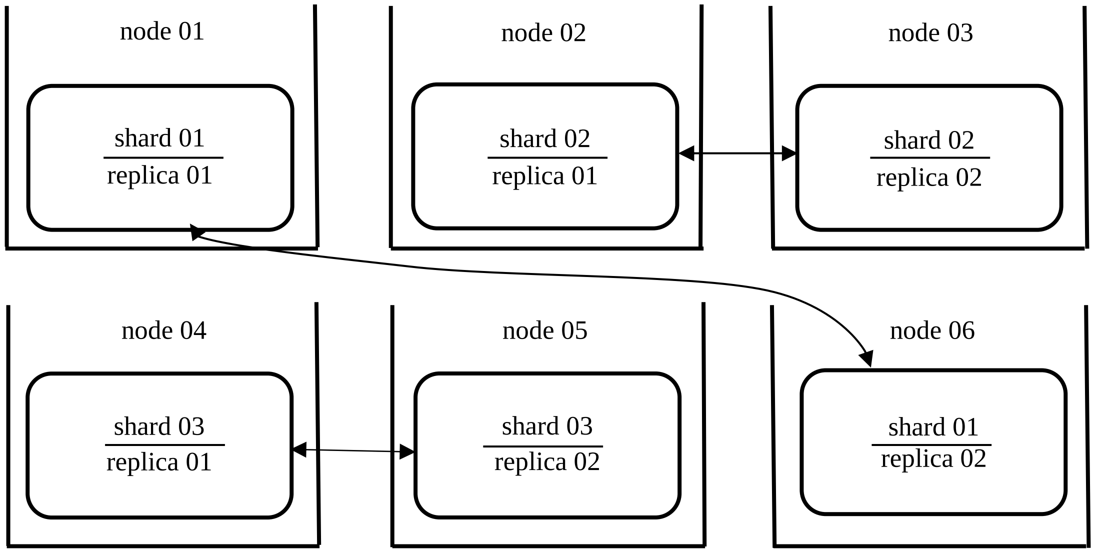
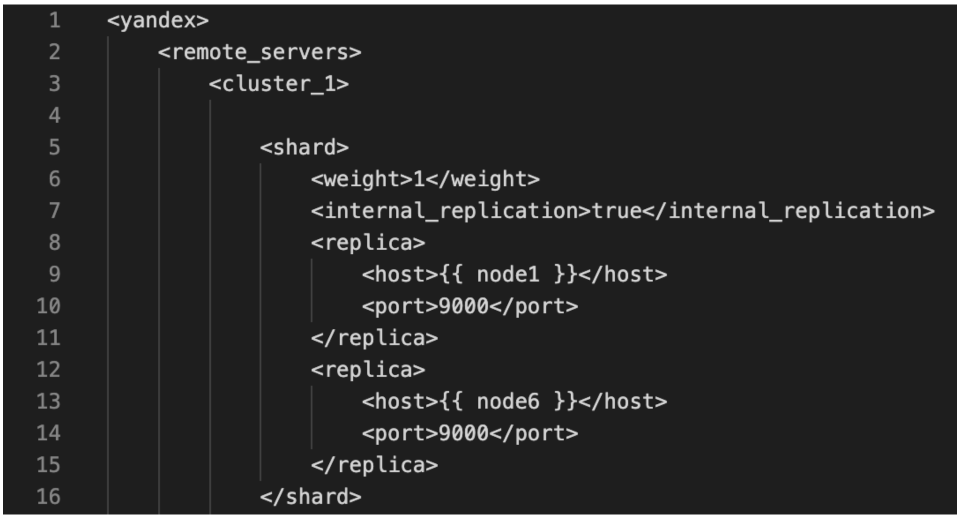


Fig. 1. Fault-tolerant cluster of 6 nodes (the authors' figure)

To replicate data and execute distributed DDL queries, we need to use +1 node with ZooKeeper installed. You can also use ClickHouse Keeper, compatible with ZooKeeper, which does not require installation on a separate server.

An example of a fragment of the configuration file is shown in Figure 2, from which it can be seen that the shard has replication configured for the 1st and 6th nodes.



```
<yandex>
    <remote_servers>
        <cluster_1>

            <shard>
                <weight>1</weight>
                <internal_replication>true</internal_replication>
                <replica>
                    <host>{{ node1 }}</host>
                    <port>9000</port>
                </replica>
                <replica>
                    <host>{{ node6 }}</host>
                    <port>9000</port>
                </replica>
            </shard>
```

Fig. 2. Fragment of the configuration file for 6 nodes (the authors' figure)

An option of the cluster configuration of 3 nodes with cyclic replication is shown in Figure 3.

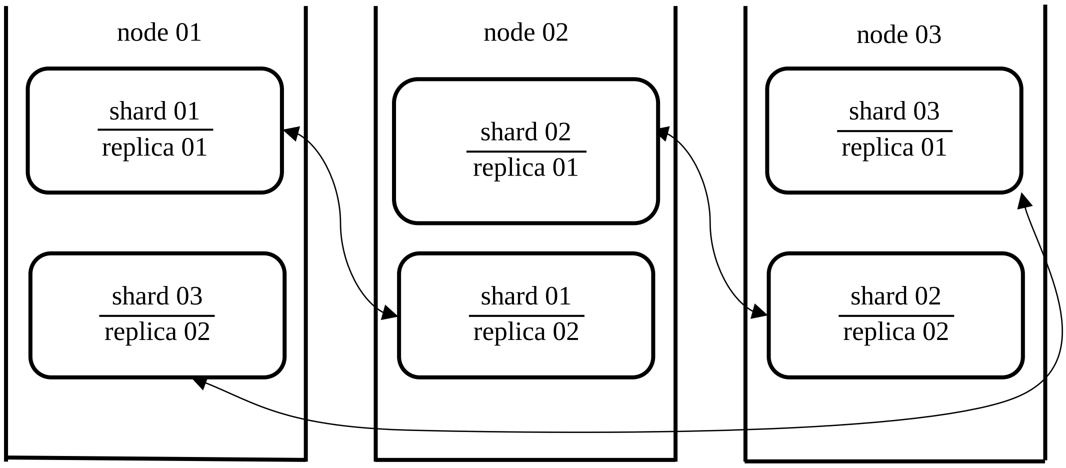


Fig. 3. Fault-tolerant cluster of 3 nodes (the authors' figure)

This implementation requires two different segments located on each node. The main problem arises due to the fact that each shard has the same table name, ClickHouse cannot distinguish one shard/replica from another when they are located on the same server.

To solve this problem, it is needed:

– to place each shard in a separate database (schema);

– to set default_database for each shard;

– to set internal_replication parameter of each shard to true;

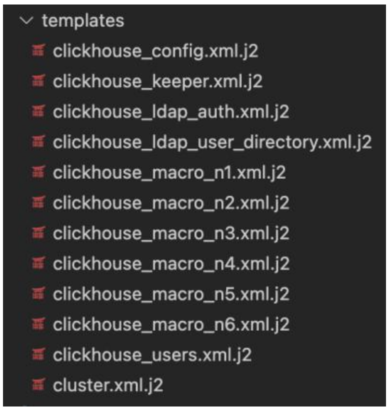
– to use an empty database parameter in a distributed table DDL script.

For this topology in an industrial environment, 6 server nodes are required, where each server stores data from only one segment, a security trapdoor for a separate database is not required. To save resources in the development or testing area, a configuration with 3 nodes can be used.

Automation is performed using Ansible Playbooks and integrated with Gitlab version control system. Thus, rapid deployment of the configuration on all nodes of the cluster is provided. When changing the configuration, it can be applied to all nodes with a single command or deploy a new DBMS cluster in a few minutes [21].

Research Results. The fault-tolerant cluster of the analytical DBMS provides redundancy for important system components, which allows for continuous operation even in case of errors in individual cluster nodes. This is done through load balancing, data replication between cluster nodes, and high reliability of the components used in the cluster. The result is an increase in the availability and reliability of the analytical DBMS, which is business-critical when analytical queries play a key role. The fault-tolerant cluster configuration of the data warehouse for analytical queries in the banking sector, taking into account the automation of the deployment process, enables to increase the reliability of the analytical data warehouse and meet the requirements for scalability. The developed task of automating cluster deployment using the mechanism of templating configuration files in Ansible Playbooks provides for the configuration of a ready-made cluster on new servers in a few minutes. The tasks of the template include operations to install the required packages, create the needed configuration and launch the cluster.

An example of configuration files for automatic deployment of a DBMS cluster is shown in Figure 4. The j2 extension says that they are created using the Jinja template engine. Purpose-built placeholders in the template provide writing code similar to Python syntax. Parameters are passed to the template for automatic insertion into the final document, thereby achieving automatic assembly into development, testing and industrial operation zones, which does not require manual modification of configuration files.



```
templates
  clickhouse_config.xml.j2
  clickhouse_keeper.xml.j2
  clickhouse_ldap_auth.xml.j2
  clickhouse_ldap_user_directory.xml.j2
  clickhouse_macro_n1.xml.j2
  clickhouse_macro_n2.xml.j2
  clickhouse_macro_n3.xml.j2
  clickhouse_macro_n4.xml.j2
  clickhouse_macro_n5.xml.j2
  clickhouse_macro_n6.xml.j2
  clickhouse_users.xml.j2
  cluster.xml.j2
```

Fig. 4. Configuration files

Description of configuration files:

clickhouse_config.xml.j2 — general cluster configuration;

clickhouse_keeper.xml.j2 — zookeeper configuration, which is responsible for node synchronization and replication;

clickhouse_ldap_auth.xml.j2 — LDAP connection configuration for data security;

clickhouse_ldap_user_directory.xml.j2 — role-based configuration by access groups to ensure data security;

clickhouse_macro_n1(6).xml.j2 — macro files (each node has its own);

clickhouse_users.xml.j2 — configuration file for creating local users needed for administration;

cluster.xml.j2 — cluster configuration file.

To test the reliability of this configuration, an experiment was conducted during which data was loaded into a DBMS cluster with a replication factor equal to 2. The dwh schemas and cluster_test_data tables were created on each of the nodes of the DBMS cluster, and a distributed table was created on the dwh cluster.cluster_test_data_distributed. The rows of the dwh.test_data_distributed table distributed across the cluster were 27,547,855. The rows of the dwh.cluster_test_data table with each of the cluster nodes are listed below:

9,186,544 rows — 1st node;

9,182,959 rows — 2nd node;

9,182,959 rows — 3rd node;

9,178,352 rows — 4th node;

9,178,352 rows — 5th node;

9,186,544 rows — 6th node.

Conspicuously, the table was distributed over the entire cluster. According to the configuration shown in Figure 1, the replication factor was 2, which means that each data block would be presented on 2 nodes. This can be seen from the number of rows on the nodes: the sixth node stored a copy of the first, the third — a copy of the second, the fifth — a copy of the fourth.

The fault tolerance of this configuration can be checked by alternately disabling nodes in the cluster. To do this, you can turn off the node or stop services on one of the nodes with the systemctl stop clickhouse-server command. During the experiment, DBMS services were stopped on cluster nodes.

With simultaneous disconnection of the 3rd, 4th, 6th or 1st, 2nd, 5th nodes that contained replicas, users continued to receive data from the dwh.cluster_test_data_distributed table, and the number of rows was equal to 27,547,855. When one of the nodes was disabled, the data continued to be displayed, and the number of rows was equal to 27,547,855. When the nodes containing the replica and the original data were disconnected at the same time, data loss occurred. This configuration can be scaled to 12 nodes, then the replication coefficient will be 3, and the sharding coefficient will be 6.

Discussion and Conclusions. The proposed solution can increase the speed of execution of analytical queries, solve problems with the scalability and reliability of data storage in banking organizations. The authors have automated cluster deployment by using templates in Ansible Playbooks, which provides setting up a ready-made cluster on new servers in minutes. This configuration can be scaled by increasing the number of nodes and adding them to the configuration files.

A set of characteristics that DBMS OLAP should correspond to was indicated, DBMS comparison was performed, a fault-tolerant cluster configuration of a data warehouse for analytical queries in the banking sector was proposed, automation of the configuration deployment process was performed. A similar solution is applicable for deployment on FreeBSD, Linux, macOS. The cluster configuration diagrams are given. This configuration can solve the problem of reliability and scalability, which is often found in organizations.

References

1. Sivov VV. *Data Security in the Business Analytics System*. In: Proc. IV All-Russian Sci.-Pract. Conference with international participation “Information Systems and Technologies in Modeling and Control”. 2019. P. 142–145.

2. Solomon Negash, Paul Gray. Business Intelligence. In: *Handbook on Decision Support Systems 2*. Springer, Berlin, Heidelberg; 2008. P. 175–193.

3. Imhoff C, Galemmo N, Geiger JG. *Mastering Data Warehouse Design: Relational and Dimensional Techniques.* John Wiley & Sons; 2003. 456 p.

4. Hugh J Watson. Tutorial: Business Intelligence – Past, Present, and Future. *Communications of the Association for Information Systems*. 2009;25:39. https://doi.org/10.17705/1CAIS.02539

5. Roscoe Hightower, Mohammad Shariat. Conceptualizing Business Intelligence Architecture. *Marketing Management Journal.* 2007;17:40–46.

6. Inmon WH. *Building the Data Warehouse*, 4th ed. John Wiley & Sons; 2005. 576 p.

7. Bogatyrev VA, Bogatyrev SV, Bogatyrev AV. Timely Redundant Service of Requests by a Sequence of Cluster. *CEUR Workshop Proceedings.* 2020;2590:1–12.

8. Henning Baars, Hans-George Kemper. Management Support with Structured and Unstructured Data — An Integrated Business Intelligence Framework. *Information Systems Management*. 2008;25:132–148.

9. Rachmiel AG, Morgan NP, Danielewski D. *Batch Management of Metadata in a Business Intelligence Architecture.* U.S. Patent No. 8,073,863 B2. 2011.

10. Dehne F, Eavis T, Rau-Chaplin A. The cgmCUBE Project: Optimizing Parallel Data Cube Generation for ROLAP. *Distributed and Parallel Databases.* 2006;19:29–62.

11. Bogatyrev V, Bogatyrev S, Bogatyrev A. Timely Redundant Service of Requests by a Sequence of Cluster. *CEUR Workshop Proceedings*. 2020;2590:1–12.

12. Milenin EI, Sivov VV. Simulation Model of Information Interaction of Measuring Devices in an Automated Environmental Monitoring System Based on IoT Technologies. *CEUR Workshop Proceedings.* 2021;2834:484–492.

13. Bogatyrev VA, Bogatyrev SV, Golubev IYu. Optimization and the Process of Task Distribution between Computer System Clusters. *Automatic Control and Computer Sciences.* 2012;46:103–111.

14. Cuzzocrea A, Il-Yeol Song, Davis KC. *Analytics over Large-Scale Multidimensional Data: The Big Data Revolution!* In: Proc. DOLAP 2011, ACM 14th International Workshop on Data Warehousing and OLAP. 2011. P. 101–104. http://dx.doi.org/10.1145/2064676.2064695

15. Sivov VV. *Sravnenie klyuchevykh programmnykh produktov dlya biznes-analitiki v bankovskoi sfere.* In: Proc. VI Int. Sci.-Pract. Conf. “Informatsionnye sistemy i tekhnologii v modelirovanii i upravlenii”. 2021. P. 281–287. (In Russ.)

16. Cuzzocrea A, Bertino E. Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach. *Journal of Computer and System Sciences.* 2011;77:965–987. http://dx.doi.org/10.1016/j.jcss.2011.02.004

17. Cattell R. Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record.* 2010;12:12–27. https://doi.org/10.1145/1978915.1978919

18. Turban E, Sharda R, Delen D, et al. *Decision Support and Business Intelligence Systems*, 9th ed. Pearson College Div; 2010. 696 p.

19. Olszak CM, Ziemba E. Approach to Building and Implementing Business Intelligence Systems. *Interdisciplinary Journal of Information, Knowledge, and Management.* 2007;2:135–148. http://dx.doi.org/10.28945/105

20. Sarawagi S, Agrawal R, Megiddo N. *Discovery-Driven Exploration of OLAP Data Cubes.* In: Proc. Int. Conf. on Extending Database Technology – EDBT’ 1998. Berlin: Springer, Berlin, Heidelberg; 1998. P. 168–182.

21. Anandarajan M, Anandarajan A, Srinivasan CA. (eds.) *Business Intelligence Techniques. A Perspective from Accounting and Finance.* Berlin: Springer-Verlag Berlin; 2004. 268 p.

About the Authors:

Victor V Sivov, postgraduate of the Computer Science Department, ITMO University (49, Kronverksky Pr., St. Petersburg, 197101, RF), ORCID, v.sivov777@gmail.com

Vladimir A Bogatyrev, professor of the Computer Science Department, ITMO University (49, Kronverksky Pr., St. Petersburg, 197101, RF), professor of the Information Systems Security Department, State University of Aerospace Instrumentation (67, Bolshaya Morskaya St., Saint Petersburg, 190000, RF), Dr.Sci. (Eng.), ORCID, ScopusID

Claimed contributorship:

VV Sivov: basic concept formulation; research objectives and tasks; computational analysis; analysis of the research results; formulation of conclusions. VA Bogatyrev: academic advising; analysis of research results; revision of the text; correction of the conclusions.

Received 01.02.2023.
Revised 17.02.2023.
Accepted 20.02.2023.

Conflict of interest statement

The authors do not have any conflict of interest.

All authors have read and approved the final manuscript.

Об авторах:

Виктор Валерьевич Сивов, аспирант кафедры «Вычислительная техника» Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (197101, РФ, г. Санкт-Петербург, Кронверкский проспект, д. 49), ORCID, v.sivov777@gmail.com

Владимир Анатольевич Богатырев, доктор технических наук, профессор кафедры «Вычислительная техника» Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (197101, РФ, г. Санкт-Петербург, Кронверкский проспект, д. 49), профессор кафедры «Информационная безопасность» Санкт-Петербургского государственного университета аэрокосмического приборостроения (190000, РФ, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А), ORCID, ScopusID

Заявленный вклад соавторов:

В.В. Сивов — формирование основной концепции, цели и задачи исследования, проведение расчетов, подготовка текста, анализ результатов исследований, формирование выводов. В.А. Богатырев — научное руководство, анализ результатов исследований, доработка текста, корректировка выводов.

Поступила в редакцию 01.02.2023.
Поступила после рецензирования 17.02.2023.
Принята к публикации 20.02.2023.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.